Connect 4

Top file: Block Diagram

P1shades(5:0) and P2Shades(5:0) allow the user to change the color of their pieces other than the standard red and blue.

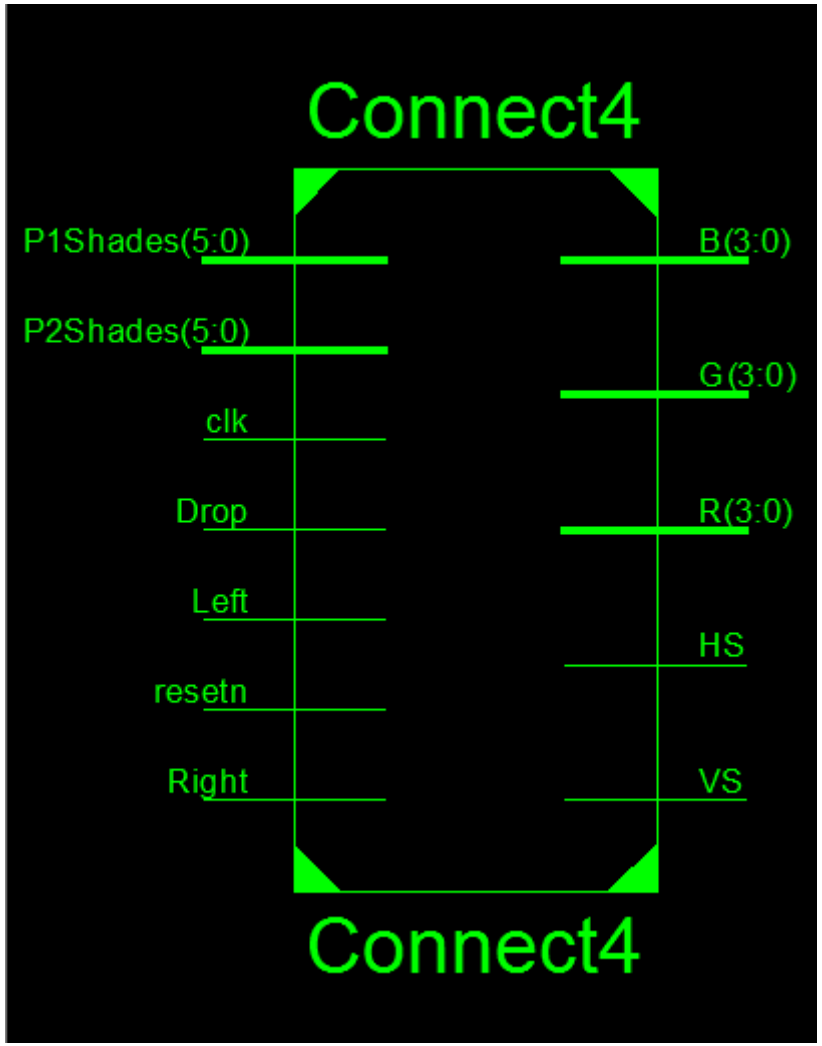
Resetn- common anode button (used for resetting)

Clk- clock set at 100 Mhz

Right, left and Drop are all buttons on the Nexys4. Right and left serve as Index selector inputs that allow the user to see where he/she is about to drop there piece.

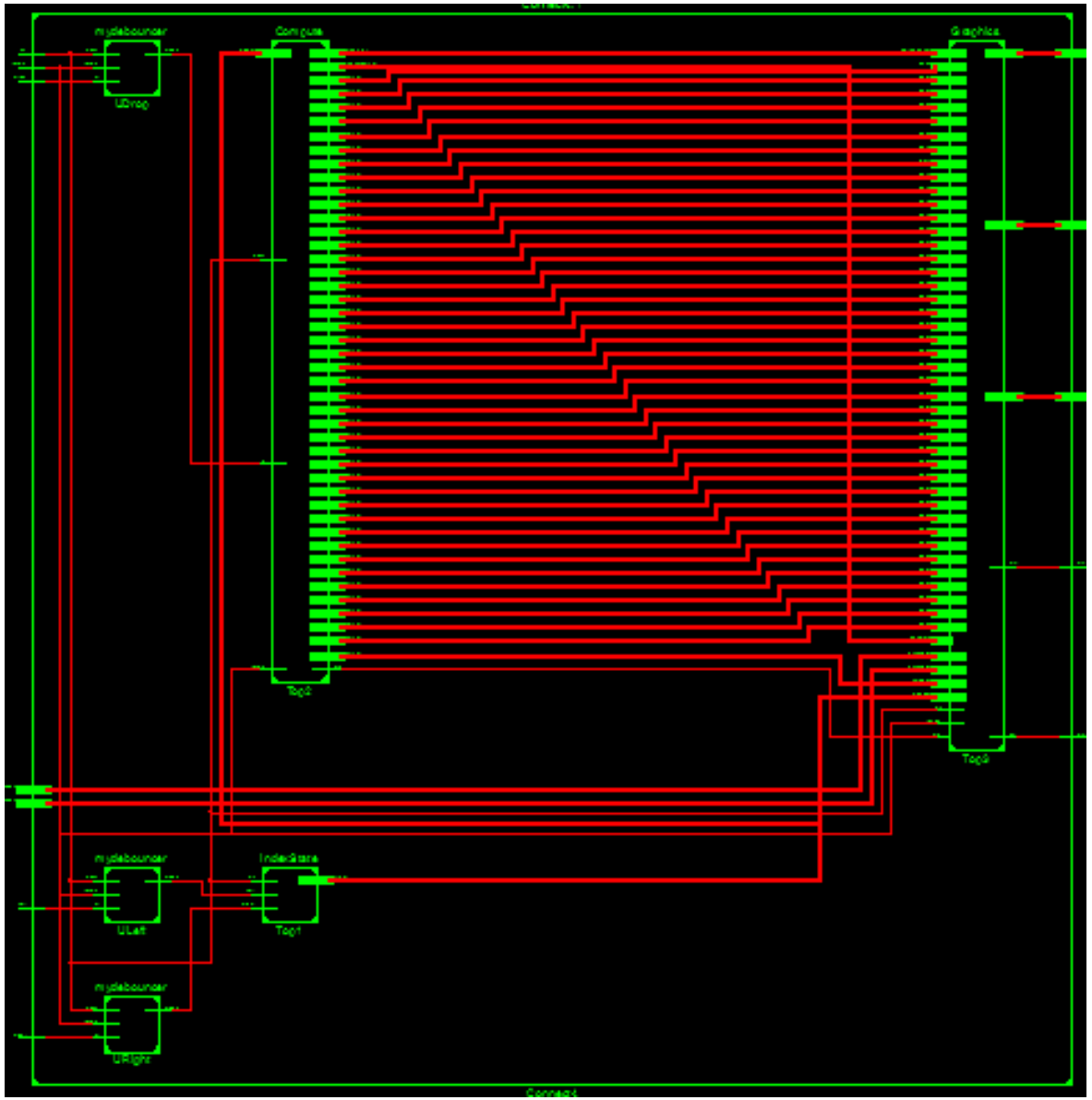
Drop places the users Indexed piece into place.

R,G,B, HS and VS are output data signals to the VGA screen



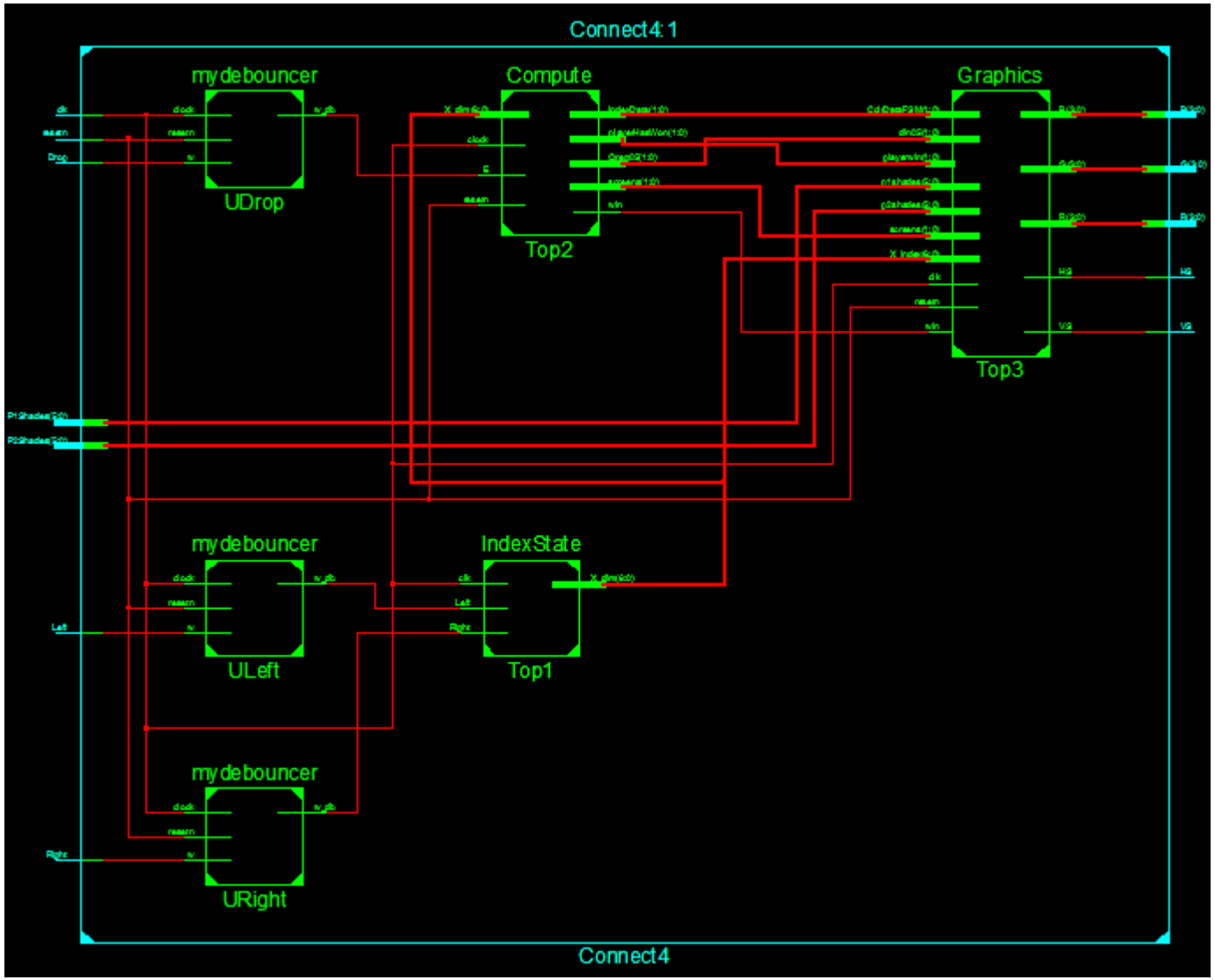
Actual Datapath Circuit

- 42 output signals from the 6x7 array of storage registers are used as inputs to the RowSelector component and the winner component. The data content within these registers is also output to the Graphics component which displays the data onto the Vga monitor.



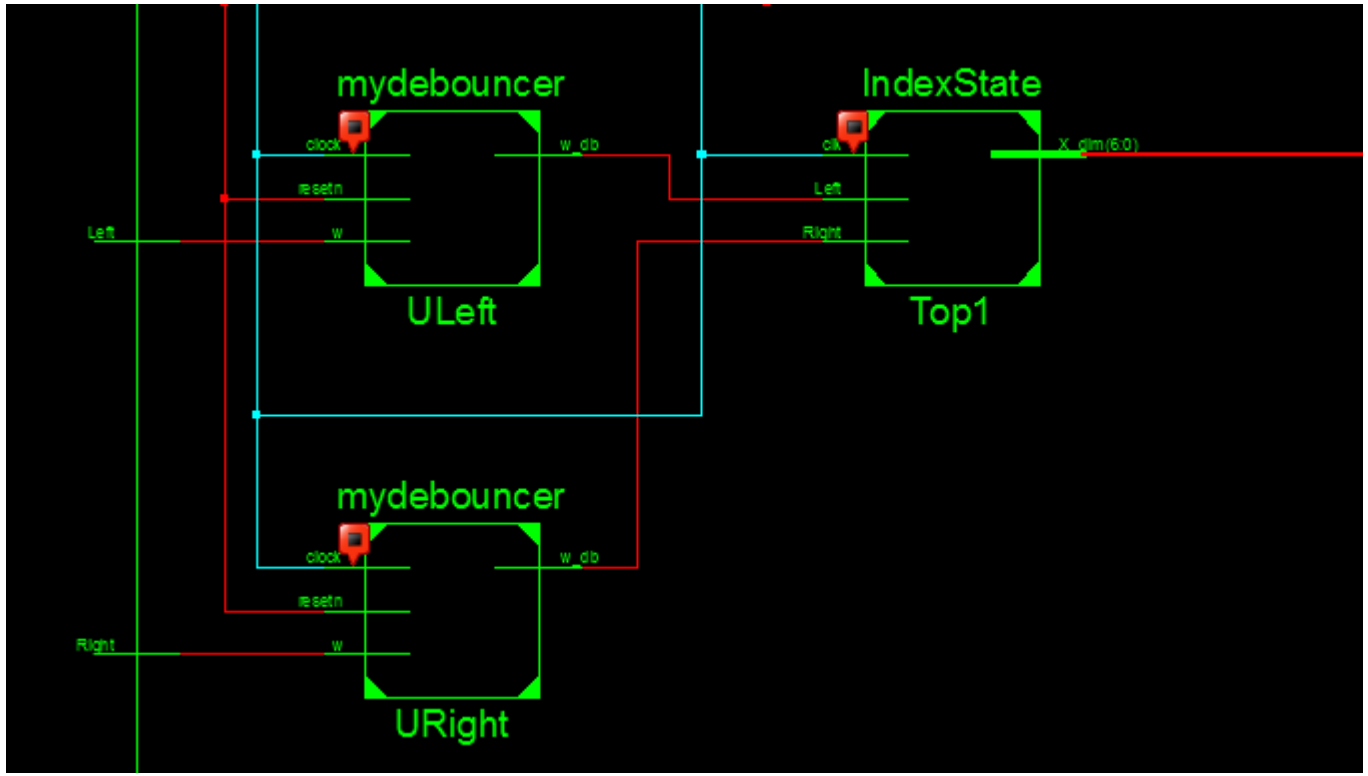
Simplified Datapath Circuit

- This circuit only contains one register. The original contains 42 separate output signals to all of the components in the Compute file.
- This image is just for demonstrative purposes.

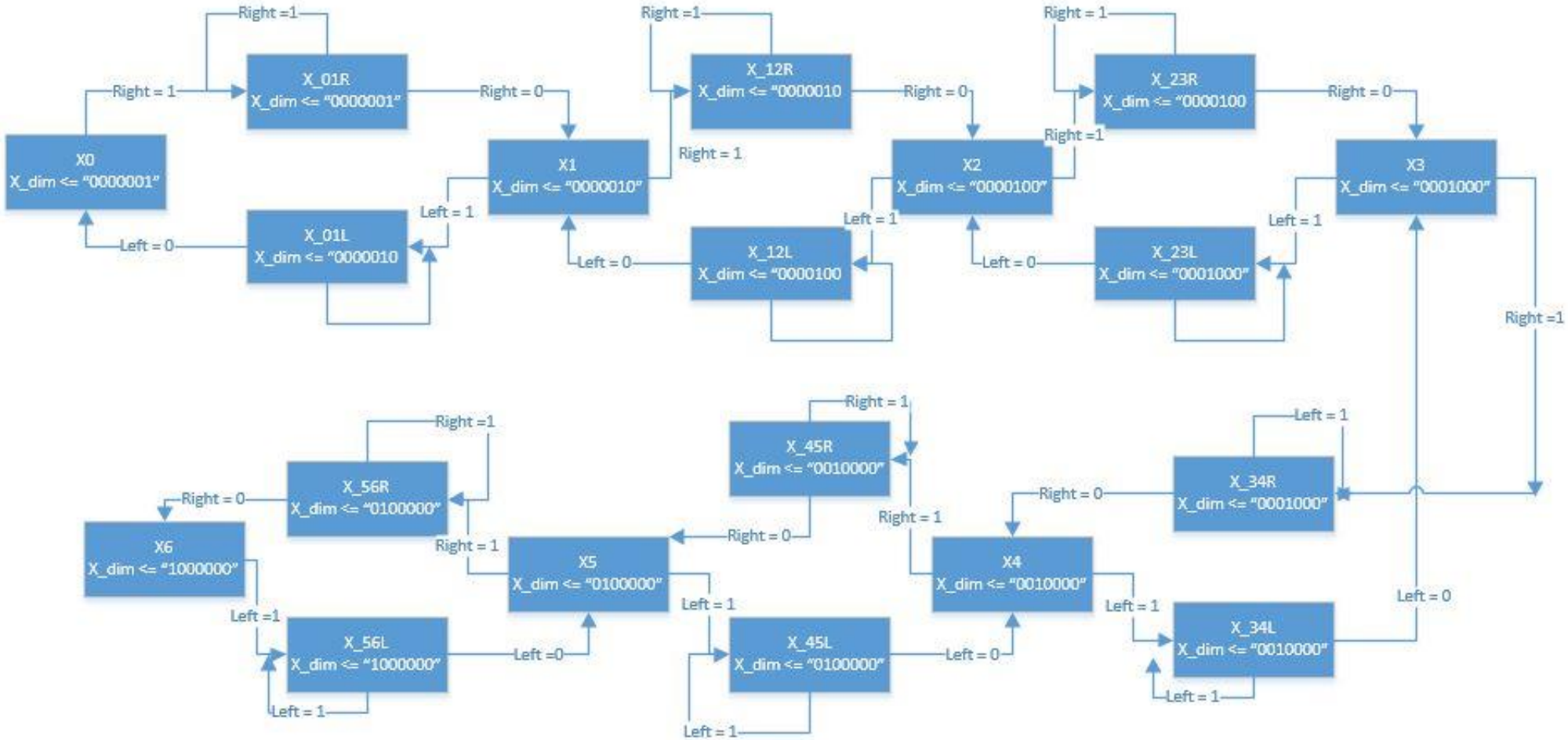


Debouncers with IndexState datapath

Left and right buttons
Are smoothened out
To get rid of mechanical
Bouncing, they are then
Fed into the IndexState
Component which acts
As a decoder to represent
Each index position above the
Connect 4 game board.

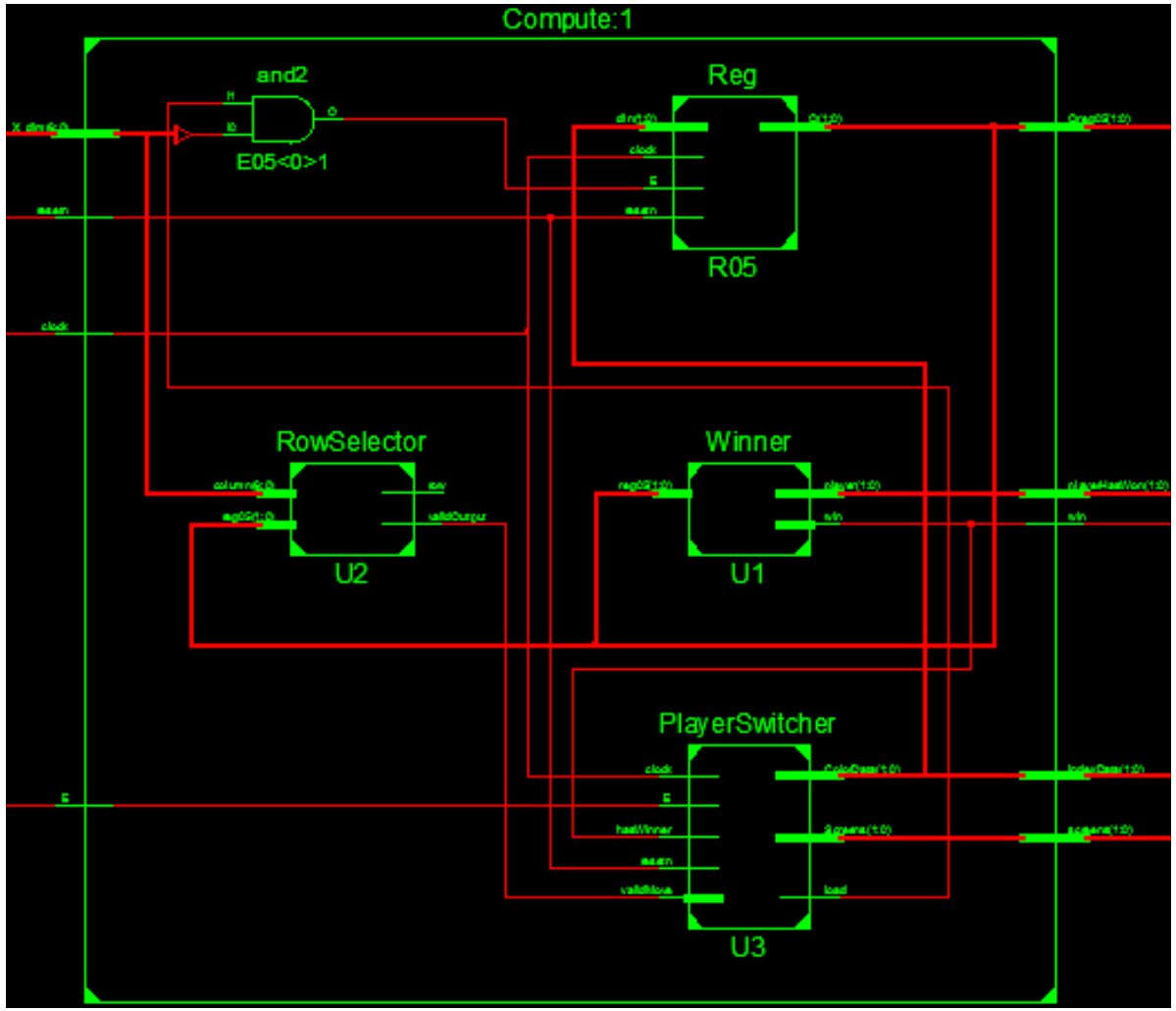


X_index Selector

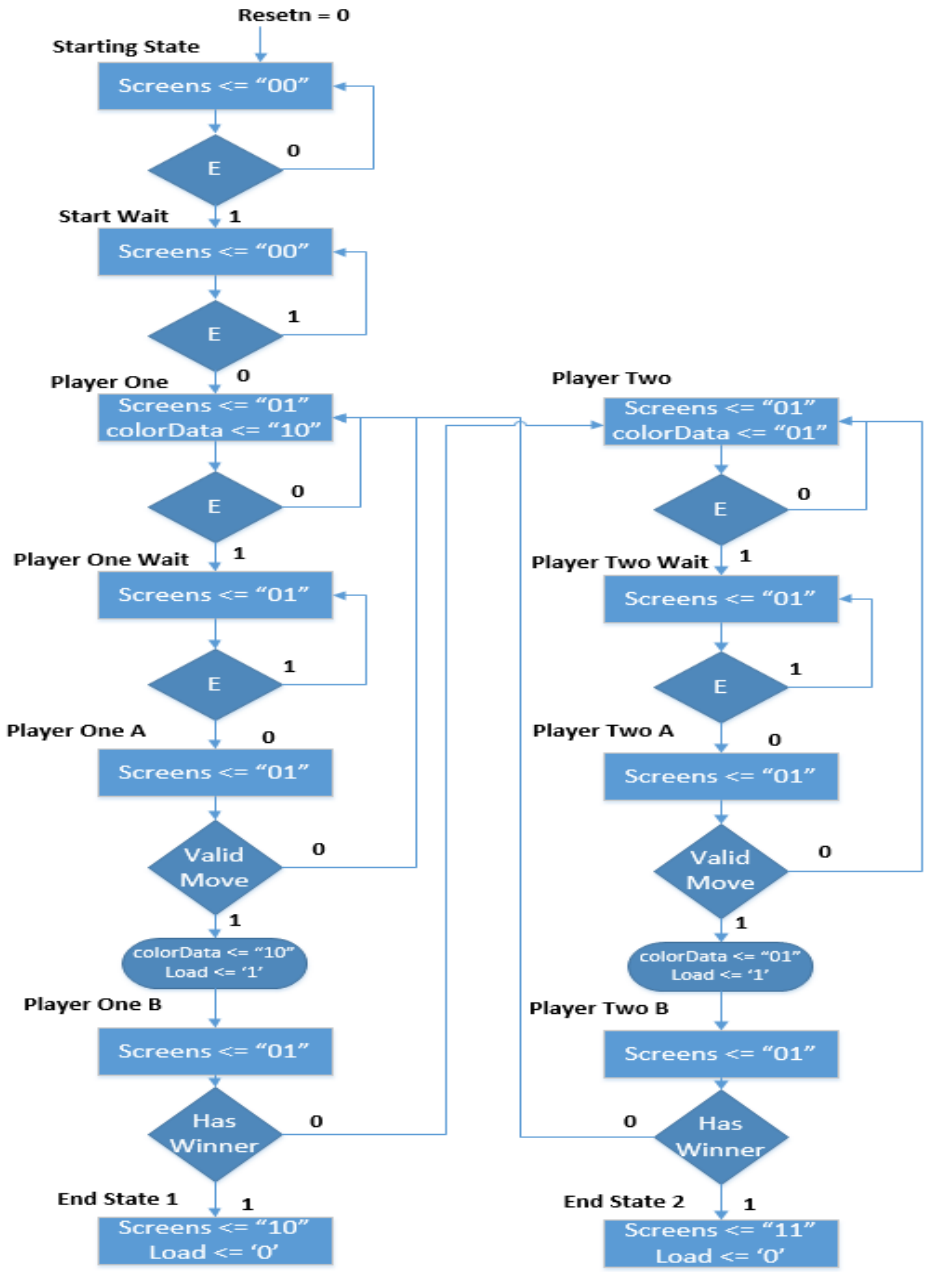


Index Selector State Diagram

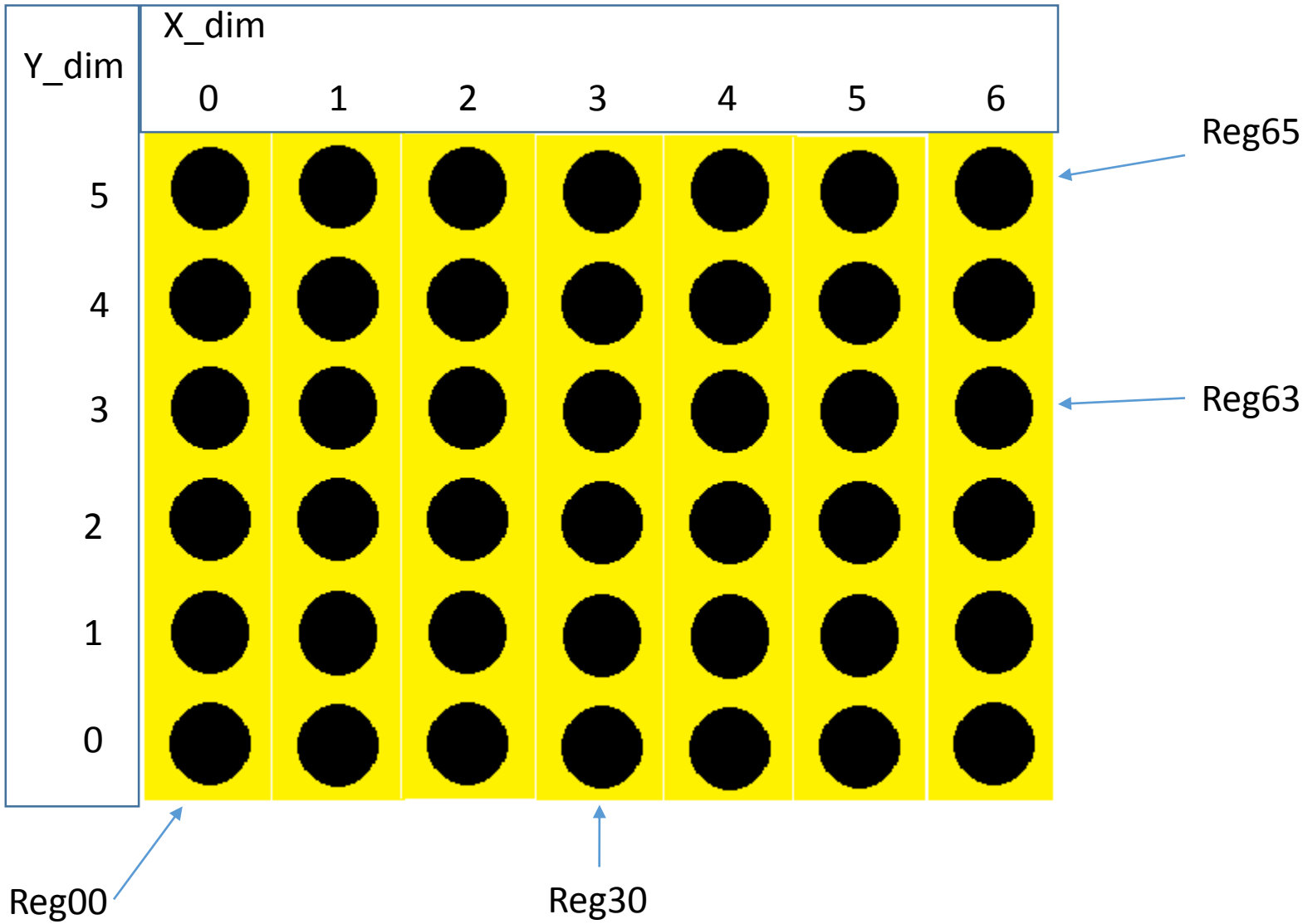
Compute: Datapath circuit Simplified



PlayerSwitcher State Machine

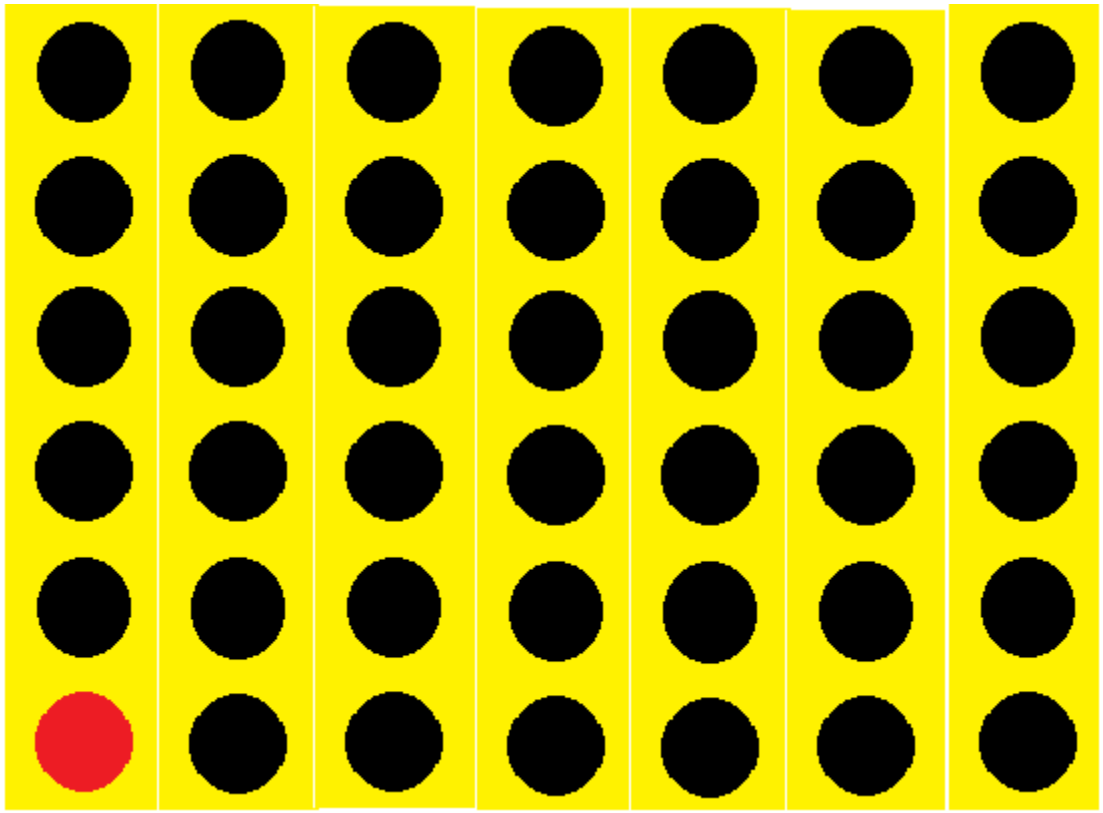


The 6x7 Grid of Registers and their Naming Convention

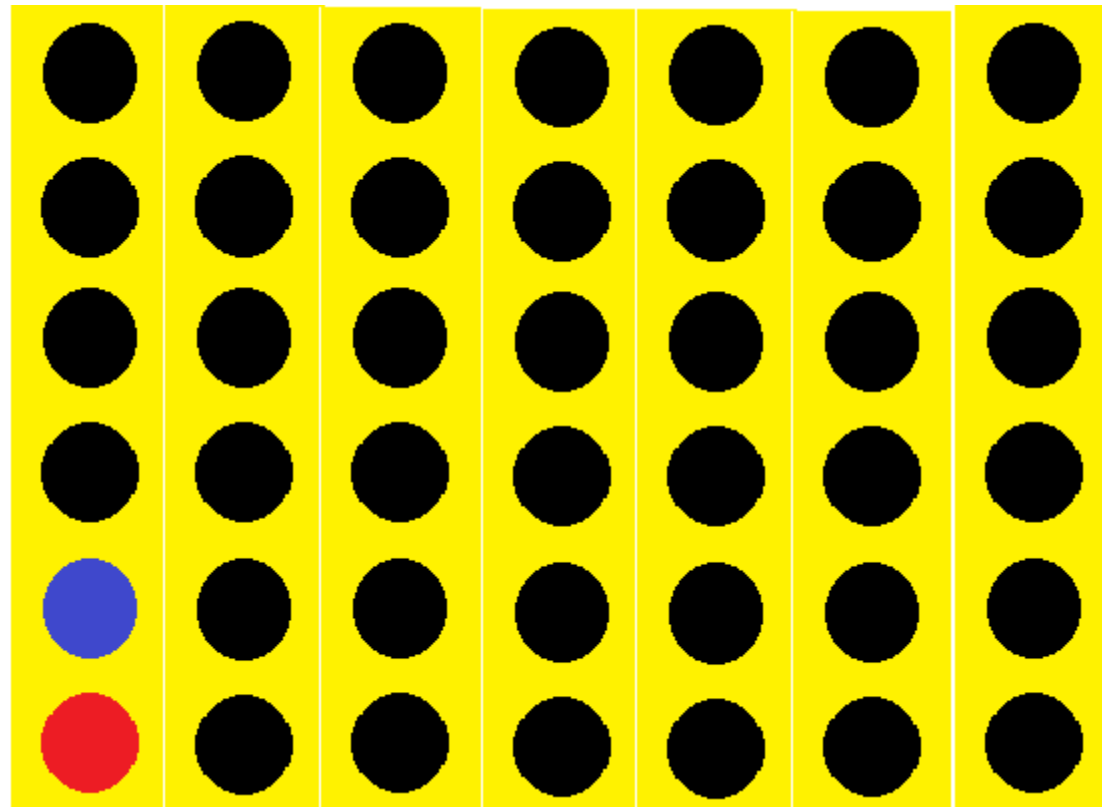


End to end Test done by the Compute (TestBench Included)

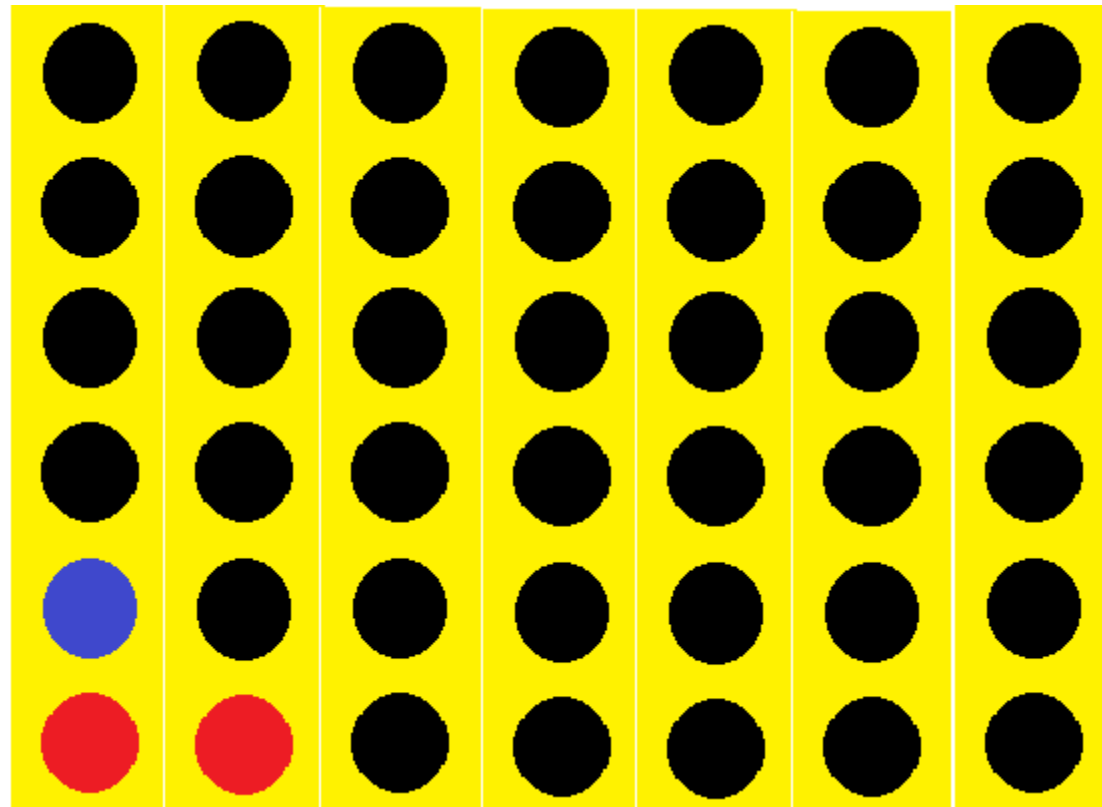
Move 1



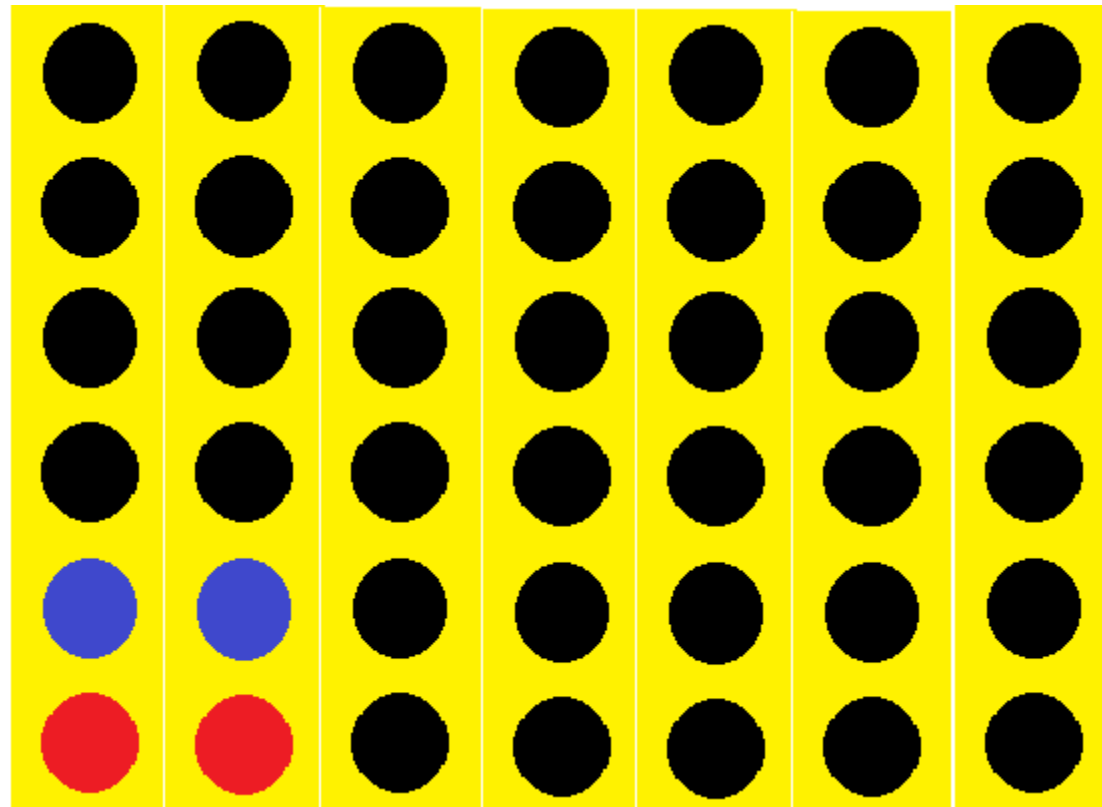
Move 2



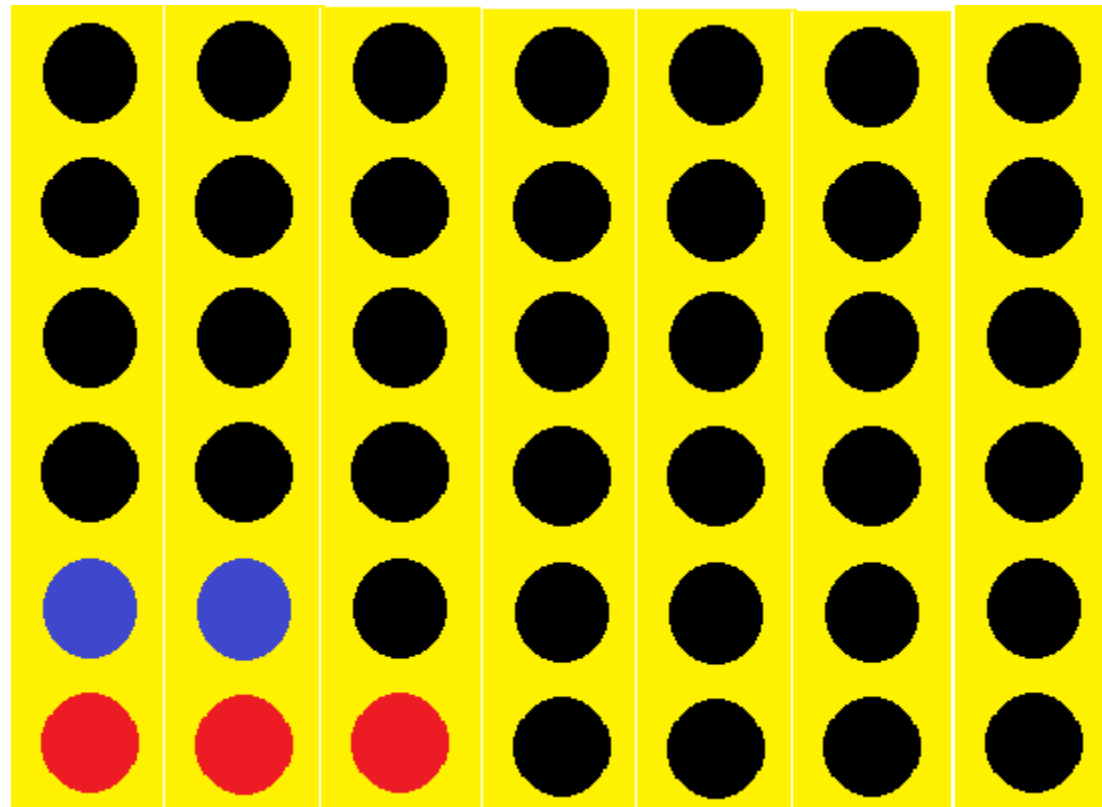
Move 3



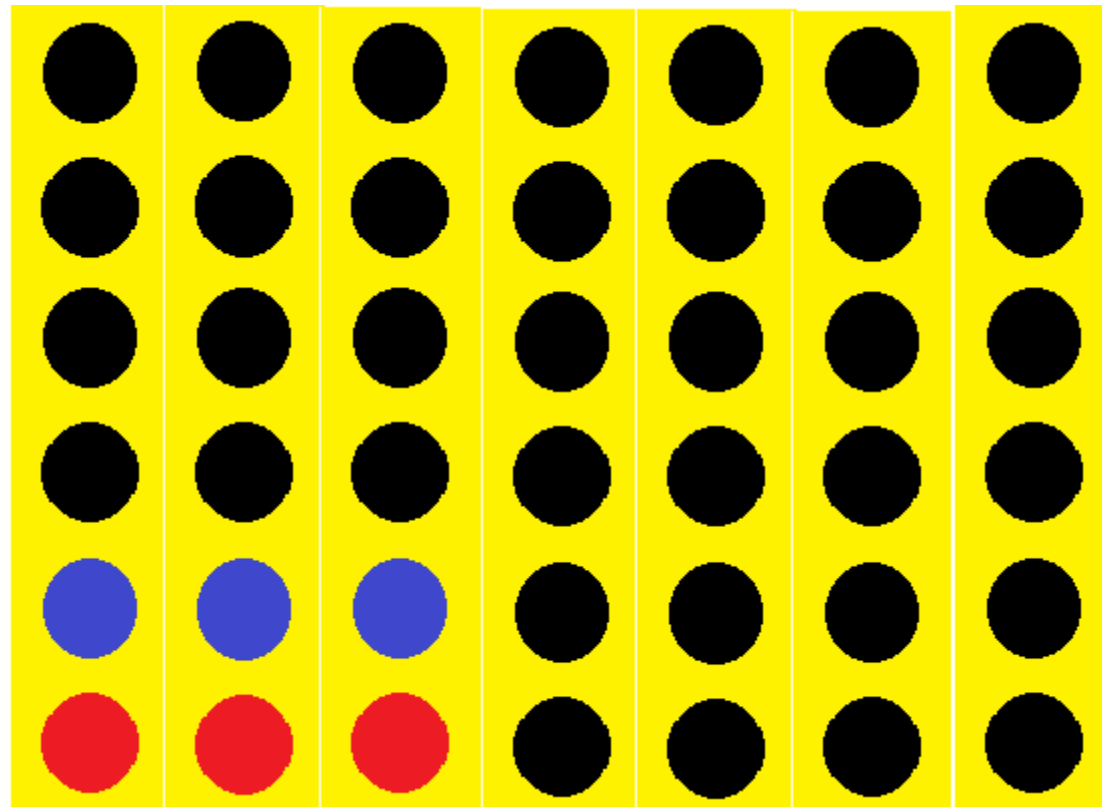
Move 4



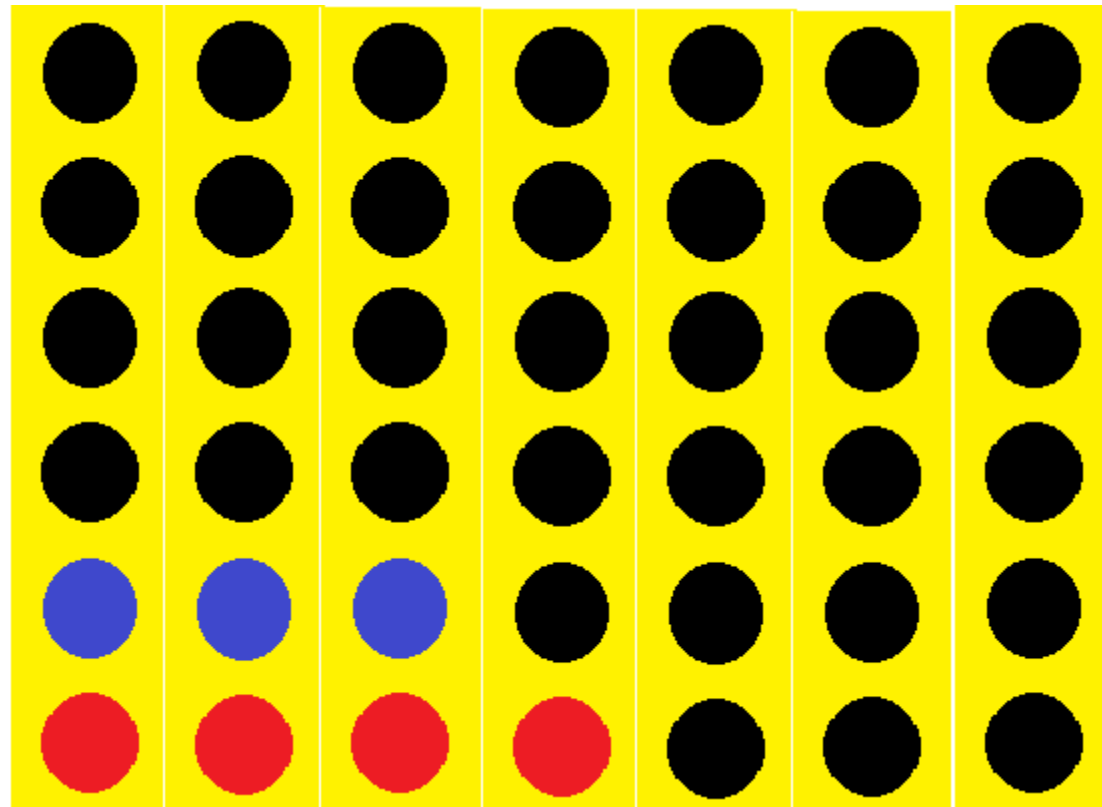
Move 5



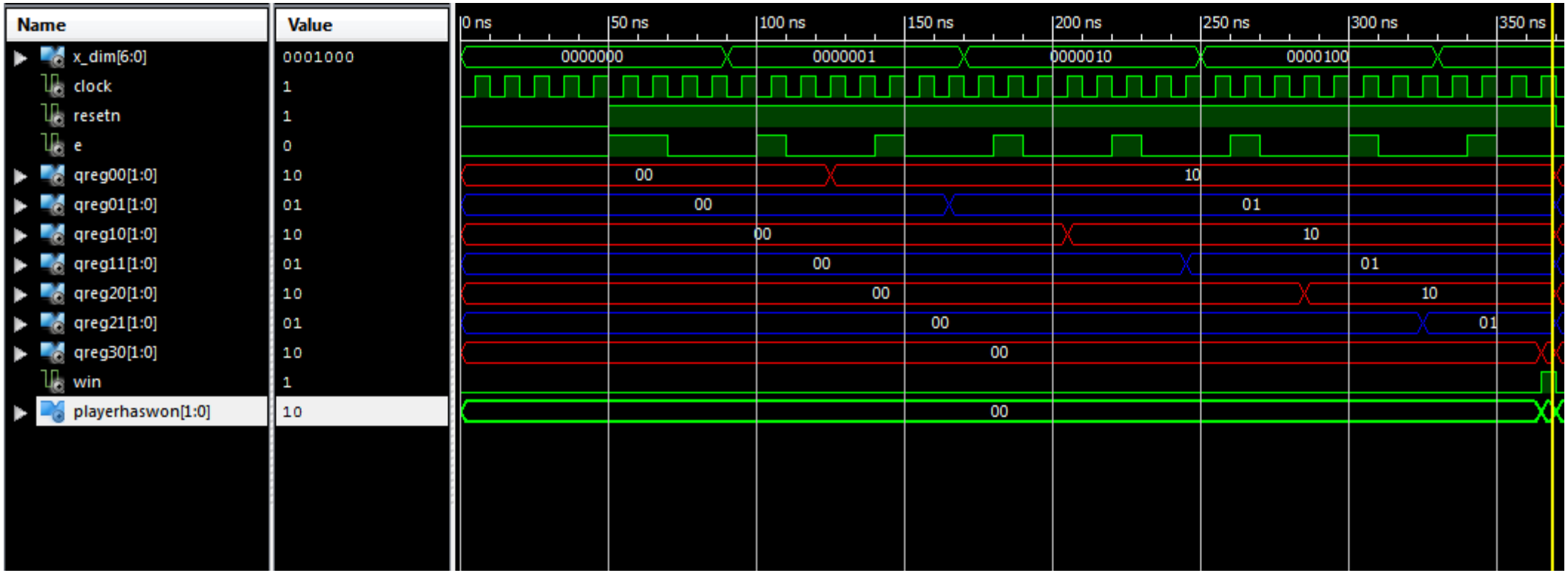
Move 6



Move 7



Timing Simulation Depicting each of the 7 turns and their corresponding values of “red=10” and “blue=01” being saved into each register.



TestBench Code

```
begin
    -- hold reset state for 100 ns.
    resetn<= '0';
    wait for 50 ns;

    resetn<= '1';
    E<= '1'; wait for T*2; E<= '0'; wait for T*2; -- exiting start_state


    --player ones turn.
    X_dim<= "0000001"; wait for T;
    E <= '1'; wait for T;
    E<= '0'; wait for T*2;

    --player two turn
    X_dim<= "0000001";wait for T;
    E <= '1'; wait for T;
    E<= '0'; wait for T*2;

    --player ones turn.
    X_dim<= "0000010"; wait for T;
    E <= '1'; wait for T;
    E<= '0'; wait for T*2;


    --player two turn
    X_dim<= "0000010"; wait for T;
    E <= '1'; wait for T;
    E<= '0'; wait for T*2;

    --player ones turn.
    X_dim<= "0000100"; wait for T;
    E <= '1'; wait for T;
    E<= '0'; wait for T*2;

    --player two turn
    X_dim<= "0000100"; wait for T;
    E <= '1'; wait for T;
    E<= '0'; wait for T*2;

    --player one wins
    X_dim<= "0001000"; wait for T;
    E <= '1'; wait for T;
    E<= '0'; wait for T*2;


end process;
```

-Reset is pressed, it instantiates all the registers with values of “00”

-Player one goes in the Left bottom corner

-Player two goes above his last move.

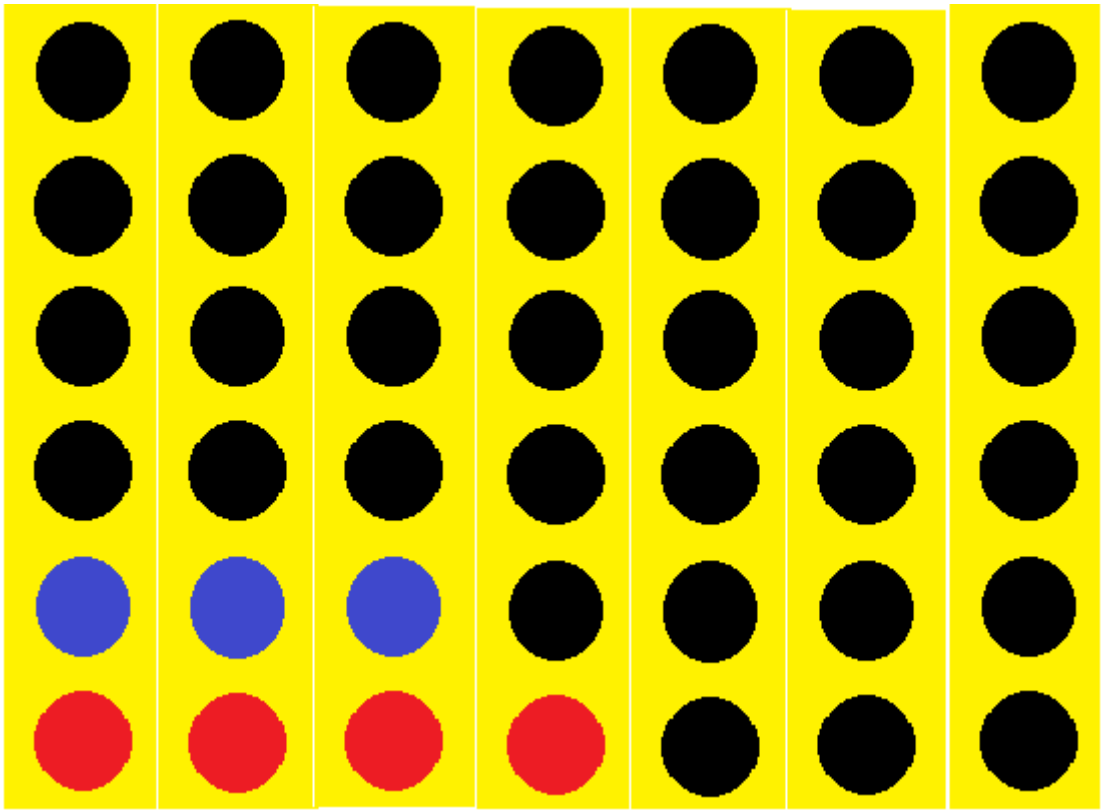
-Player one goes one index to the right.

-Player two goes above his last move.

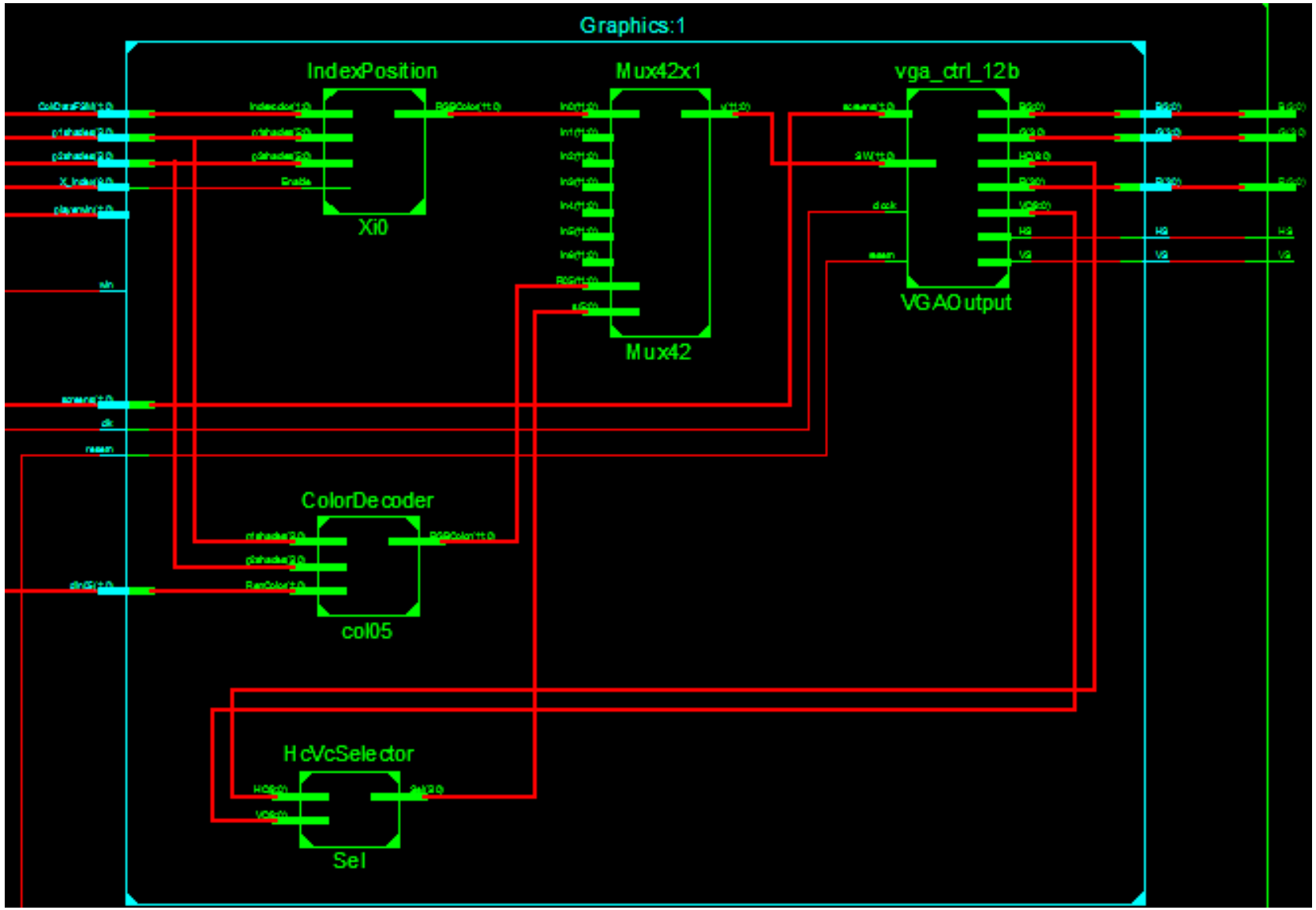
-Player one goes one index to the right.

-Player two goes above his last move.

-Player one goes one index to the right.



Graphics: Datapath Circuit Simplified



This circuit takes 49 inputs total.
42 inputs from the register contents
Containing a specific value for a
Color: red=10 and blue=01.
The value is then decoded into
12 bit RGB and used as an input
To a 42x1 Multiplexor.

The multiplexor's selector is based
Upon HC and VC values which are
Output from the VGA_ctrl_12b.
HC and VC give reference to the
X and y coordinates on the VGA
Screen.

The VGA_ctrl_12b then outputs R,G,
B, Hs and V to a Vga monitor.

That’s it.

2 volunteers to play?